

What is claimed is:

1. A method performed by a computer system to process an extensible markup language input stream using discrete software components mapped to tags contained in the input stream, comprising:
 - parsing the input stream,
 - as a tag is parsed, invoking the software component mapped to it.
2. The method of claim 1, wherein the extensible markup language is XML.
3. The method of claim 1, wherein the discrete software components are classes in an object-oriented programming language or procedures or functions in a procedural programming language.
4. The method of claim 3, wherein the input stream comprises at least one tag which is formed by a start-tag and an end-tag, and wherein the discrete software components comprise at least one of a method which is invoked as the start tag is parsed (init method) and a method which is invoked when all children of the tag have been parsed (run method).
5. The method of claim 1, wherein, when the input stream is parsed, a hierarchical memory structure, which corresponds to the input stream, is also built in a memory of the computer system.
6. The method of claim 1, wherein the mapping between the tags and the discrete software objects is changed before, during or after the parsing process.

1 7. The method of claim 1, wherein an extensible markup language output
2 stream is generated, and the output stream is used as an input stream for another
3 execution of the method.

4
5 8. The method of claim 1, wherein the computer system is a server in a
6 network, and the input stream processed by the server is comprised in a request
7 received from a client over the network or is comprised in an output from a database.

8
9 9. The method of claim 1, wherein the invoked software components
10 comprise at least one software component for accessing a database.

11
12 10. The method of claim 1, wherein the invoked software components
13 comprise at least one of a software component for sending electronic mail and a
14 software component for sending facsimiles.

15
16 11. The method of claim 1, wherein the computer system comprises a local
17 computer and a remote computer which communicate with each other, and at least
18 some of the commands given to the local computer are executed on the remote
19 computer, but the results of the execution are output on the local computer so that the
20 commands appear to be executed locally, wherein the communication between the
21 local and the remote computers comprises a extensible markup language stream,
22 wherein the method is performed by at least one of the remote computer and the local
23 computer so as to execute the commands or output the results of the execution.

24
25 12. A computer system, comprising:
26 a processing unit and storage for processing programs,
27 bindings representing a mapping between tags and discrete software
28 components,

1 a software engine comprising:

- 2 - a reader component that reads an extensible markup language input
3 stream containing at least one tag,
4 - a parser component that parses the input stream,
5 - an execution component that, as a tag is parsed, invokes the software
6 component mapped to the tag.

7
8 13. The computer system of claim 12, which comprises two or more of the
9 software engines, wherein at least one of the software engines generates an
10 extensible markup language output stream, and the output stream is used as an input
11 extensible markup language stream for another one of the software engines.

12
13 14. The computer system of claim 12, which is a server in a network, and
14 the extensible markup language input stream processed by the server is comprised in
15 a request received from a user over the network or is comprised in an output from a
16 database.

17
18 15. The computer system of claim 12, which comprises a database, and
19 wherein the invoked software components comprise at least one software component
20 for accessing the database.

21
22 16. The computer system of claim 12, which comprises at least one of an
23 email dispatch system and a facsimile dispatch system, wherein the invoked software
24 components comprise a software component for sending electronic mail or a
25 software component for sending facsimiles.

26
27 17. The computer system of claim 14, which comprises a local computer
28 and a remote computer which communicate with each other, and at least some of the
29 commands given to the local computer are executed on the remote computer, but the

1 results of the execution are output on the local computer so that the commands
2 appear to be executed locally, wherein the communication between the local and the
3 remote computers comprises an extensible markup language stream, and wherein at
4 least one of the remote and the local computers has a software engine so as to
5 execute the commands or output the results of the execution.
6

7 18. A computer program product including program code for execution on a
8 computer system, said program code for processing an extensible markup language
9 input stream using discrete software components mapped to tags contained in the
10 input stream, said program code for directing the computer system to:

- 11 - parse the input stream,
- 12 - as a tag is parsed, invoke the software component mapped to it.
- 13

14 19. The computer program product of claim 18, wherein the program code
15 is stored on a computer-readable data carrier or is in the form of signals transmitted
16 over a computer network.
17

18 20. The computer program product of claim 18, wherein the program code
19 comprises the following classes:

- 20 - a class which parses the input stream,
- 21 - a class which implements a parser interface,
- 22 - a class which creates a document,
- 23 - a class which creates a taglet, i.e. which binds objects to tag names,
- 24 - a class which provides behavior for the taglets.